Фронтенд стек знань (General + React.js)

Рік створення: 2024.

Автор підбірки: https://www.linkedin.com/in/dima1popov/

Це список питань, які орієнтовані приблизно на 5 годин обговорення, щоб виявити цілісність знань Старшого (Senior) фронтенд інженера. Це не питання для стандартної співбесіди на півтори години. Список питань підібраний на основі практичного досвіду автора та аналізу ринку. Крім того, автор орієнтувався на стек знань SWEBOK (IEEE), Scrum PSD, ISTQB, низку онлайн тестів (W3Schools), вакансій та публічних співбесід. Список орієнтований на цілісного інженера, який розуміє повний цикл розробки програмного продукту. Більшість питань в списку є відкритого типу (open-ended questions), які вимагають розгорнутої відповіді.

Питання:

Хто такий Senior та Senior frontend? (IEEE, Market)
Хто такий програмний інженер? (Friedrich Ludwig Bauer definition, IEEE definition)
Хто такий фронтенд інженер?
Що таке інтерфейс користувача (GUI)?
Що таке алгоритм?
Що таке програма (чи будь-які інструкції є програмою)? (Niklaus Wirth definition of program)
Складність алгоритму? О-нотація.
Що таке складність програми?
Що таке сторі пойнти (story point)?
Що таке архітектура програми (додатку)? (R.Martin, “Clean architecture”)
Що таке програмний продукт?
Який життєвий цикл продукту (SDLC)?
Як встановлювати версії продукту? (semver.org)
Що таке методології розробки продукту (Agile, waterfall, spiral)?
Що таке Waterfall model?
Що таке Scrum? (scrum.org)
Що таке Daily meet, ціль та хто на ньому?
Що таке Scrum team?
Як визначити KPI програміста?
Що таке XP (extremeprogramming.org)?
Що таке Definition of Done and Acceptance tests?
Які методи та правила декомпозиції задачі? (YAGNI, DRY, Top-down, TDD friendly)
Що таке TDD та BDD?
Що таке Git та GitFlow?
Які Git conventions?
Що таке npm?
Що таке CI/CD? (Що таке дванадцяти факторний додаток? https://12factor.net/)
Що таке code linter?
Що таке production build?
Що таке Docker?
Що таке docker image, container, volume?
Що таке HTML?
HTML це мова програмування?
Що таке HTML семантика?
Скільки H1 та H2 заголовків може бути на одній сторінці?
Для чого тег <meta /> ?
Що таке contenteditable?
Що таке тег <canvas/> та WebGL?
Що робить тег <script /> та його атрибути defer і async?

Які є типи списків? (ol, ul, dl)
Що таке CSS?
Що таке Cascading?
Які є CSS селектори?
Яка специфічність селекторів?
Чи можна використовувати цифри та тире в назвах CSS класів?
Як центрувати блок? (Декілька способів)
Що таке flexbox?
Що робить властивість box-sizing?
Які юніти в CSS? (em, rem, cm, px, pt, %, vh, vw).
Що таке регулярний вираз?
Чи є в CSS регулярні вирази?
Що таке CSS функції? (calc(), repeat(), clamp())
Що таке CSS Media Queries?
Яка різниця між inherit та currentColor в CSS?
Які стани елемента знаєш? (hover, active, focus, disabled, pressed)
Розкажи про змінні в CSS?
Що таке SASS?
Що таке window.devicePixelRatio?
Що таке дизайн токени та магічні числа? (https://atlassian.design/tokens/design-tokens)
Що таке ECMA script?
Що таке JavaScript (JS)?
JS однопоточний чи багатопоточний? (Що таке Web Workers API?)
Що таке statement та expression в JS?
Які типи даних в JS ? (їх 8)
Що таке примітивні та посилальні (reference) типи?
Що таке event loop в JS?
Що таке promises?
Що таке Promise.all?
Що таке контекст (this) в JS?
Що таке замикання (closure) в JS?
Що таке прототип об'єкта?
Що таке чиста (pure) функція?
Що таке функція конструктор?
Що таке стрілочна (arrow) функція й чи може вона бути конструктором?
Що таке hoisting?
Що таке Temporal Dead Zone in JavaScript?
Які рушії (engines) JS знаєш? (V8, ...)
Що таке TS?
Які відмінності TS від JS? (typing, compiling, enum, tuples, interface, generics, structural typing except literals, decorators).
Що таке AssemblyScript?
Що таке функціональне програмування?
Які принципи OOP?
Що таке SOLID?
Що таке бізнес логіка (Domain-driven design)?
Що таке шаблон Декоратор?
Що таке шаблон Singleton (Одинак)?
Що таке DOM та DOMContentLoaded, onload?
Що таке Event delegation and propagation?
Що таке event bubbling?
Що таке React?
Які основні фішки React? (JSX, Virtual DOM, Hooks, One way data binding)
Які алгоритми Virtual DOM та reconciliation?
Огляд хуків. Хуки асинхронні? Хуки чисті функції?
Як працюють хуки під капотом?
Які антипатерни використання useEffect?
Які антипатерни використання useContext?
Який життєвий цикл React компонента?
Що таке React Portal?
Що таке HashRouting в React?
Чому використання тегу <a> в React є неоптимальне?

Як працювати з SVG в React?
Що таке стан (state) додатку?
Що таке Flux?
Що таке Redux? (store, reducer, action)
Як працює Redux під капотом?
Що таке createAsyncThunk in Redux?
Що таке RTK Query?
Які є види рендеренгу? (CSR, SSG, SSR)
Що таке CSS-in-JS?
Що таке utility first approach в CSS?
Що таке клієнт і сервер?
Що таке HTTP?
Що таке REST API?
Що таке GraphQL?
Що таке WebSocket?
Що таке TCP/IP?
Що таке URL?
Що таке CORS?
Що таке JWT (токен)?
Як створити login логіку в React?
Що таке HTTP Cookies?
Що таке Storage API? Яка різниця між localStorage та sessionStorage?
Браузер Dev tools? Розширення (Perfect pixel) та лайф хаки.
Що таке Storybook?
Що таке interface grid (12)?
Що таке responsive design?
Що таке accessibility? (WCAG)
Що таке паралакс?
Що таке drag and drop?
Що таке progressive enhancement?
Що таке тег, елемент, компонент та блок у фронтенді? Як вони пов'язані? (Look Atomic Design and WebComponents)
Що таке чистий код? (Назви змінних, нотація (camelCase, Hungarian notation), показники якості коду)
Які принципи рефакторингу? Ключове слово debugger в JS.
Які принципи тестування? (“Art Of Software Testing”, 3Ed, Glenford J. Myers)
Як тестувати UI? (PerfectPixel, DevTools, Snapshot testing)
Що таке безпека (XSS, CSRF, npm-audit)?
Що таке SEO?
Які шляхи покращення Performance? (culling, debounce, throttle, lazy loading, Virtual scroll, sprites for icons)
Що таке Мікрофронтенд? (Module federation)

Відповіді можна знайти в наступних джерелах

Книги

https://exploringjs.com/impatient-js/
https://eloquentjavascript.net/
https://scrumguides.org/
https://www.computer.org/education/bodies-of-knowledge/software-engineering/v3

Сайти

HTML, CSS, JS, TS
https://developer.mozilla.org/
https://www.w3schools.com/

https://www.typescriptlang.org/
https://tc39.es/ecma262/
https://html.spec.whatwg.org/
https://caniuse.com/
https://roadmap.sh/frontend
https://bradfrost.com/blog/post/atomic-web-design/
https://tailwindcss.com/

UI/UX
https://atlassian.design/tokens/design-tokens
https://www.contentful.com/blog/design-token-system/
https://lawsofux.com/

React
https://react.dev/
https://redux-toolkit.js.org/
https://nextjs.org/

Tests
https://glossary.istqb.org/en_US/
https://qaukraine.online/
https://refactoring.guru/ru/design-patterns
https://playwright.dev/
https://testing-library.com/

Agile, Scrum, XP
https://www.scrum.org/
http://www.extremeprogramming.org/
https://agilemanifesto.org/iso/uk/manifesto.html

Code style, GitFlow
https://nvie.com/posts/a-successful-git-branching-model/
https://www.conventionalcommits.org/en/v1.0.0/
https://semver.org/
https://google.github.io/styleguide/
https://git-scm.com/
https://12factor.net/

Self Assessment (Computer science)
https://github.com/dima-popov/SelfAssessment
https://www.notatki-z-informatyki.info/
https://www.geeksforgeeks.org/

Online tests
https://www.scrum.org/open-assessments/scrum-developer-open

https://www.w3schools.com/quiztest/
https://www.w3schools.com/react/
https://www.codewars.com/
https://leetcode.com/

Online courses
https://www.udemy.com/
https://www.linkedin.com/learning

Додатково ключові слова та фронтенд технології

Web frontend software engineer body of knowledge (sketch):

---Software engineer:

Requirements (Acceptance tests, Definition of done),

Estimation (T-shirt and pocker estimation, story points),

Decomposition (YAGNI, DRY, Bottom-up, TDD),

Construction (IDE, Git, dev containers),

Testing (TDD, Unit tests, E2E tests),

Debugging (time traveling debugging, node --inspect),

Maintenance (A/B Testing, logging, monitoring, CI/CD),

OOP (4 principles, SOLID),
FP (immutability, curring, pure functions, firts-class functions),

Design patterns (Iterator, Decorator, Singleton, Proxy, Observer, Prototype, Factory method, Chain of responsibilities),

Async code, threads and processes, Race condition,

Clean code (camelCase, FP principles, SOLID),

Security (XSS, CSRF, npm-audit),

Performance (culling, debounce, throttle, lazy loading, Virtual scroll, sprites for icons),

O-notation,

Binary search,

A read–eval–print loop (REPL),

Linux shell (CLI),

---Web:

HTTP protocol (HTTP 2),
HTTP status code (10*, 20*, 30*, 40*),
HTTP methods (GET, HEAD, POST, PATCH, PUT, DELETE, OPTIONS).
ContentTypes (Mime),
Cookies,
CORS,
CDN,
RestAPI,
GraphQL,
gRPC,
WebSockets,
Ajax, XMLHTTPRequest,
URI, URL,
ReadableStream,
JWT,

---Frontend:

CSS selectors,
CSS cascading,
CSS specificity,
CSS pseudo-selectors,
CSS media queries,
CSS function (calc, repeat),
CSS units (px, rem, pt, cm, vh),
Flexbox. Grid.
CSS approaches (Utility, BEM, SMACSS),
(SASS, Less),
CSS-in-JS, styled-components,

HTML tags,
HTML semantics,
SVG, WebP,
DOM,
Shadow DOM,
DOMContentLoaded, onload,
Events delegation,
Accessibility,
SEO,
Design systems (Material UI, Tailwind UI),
UI/UX (Color circle, DragAndDrop, Wizard, Occam's razor, Information architecture, Miller's Law, Tesler law, parallax, WYSIWYG),
Figma,

Browser Dev tools,
BOM,
WebComponents,
Timers,
Dev tools,
WebWorkers,
Wasm and asm.js,
WebGL,
WebRTC,
Service workers,
PWA (Battery API, Bluetooth API),
Storage API,
IndexedDB,

JS,
Variable,
Closure,
Scope,
"this" (context),
Hoisting,
First-class citizens,
Data types (7 primitives and object),
Type comparison, shallow & deep,
falsy values,
OOP in JS
Event loop,
async/await,

TS,
Interfaces,
Structural (& nominal) typing,
Enum,
Generics,
Decorators,
asserts (type guards),

React hooks,
JSX,

Virtual DOM,
Server components,
Client Routing (React Router),
Redux, Flux, State,
Portal,
ReactPDF,
Formik,
zod.js,

SPA, MVC,
SSG,
SSR (Server side rendering),
CSR,
Microfrontend (module federation),
Backend-for-frontend,

CI/CD
Docker,
Mono repository,
.gitignore,
SSH,
ESLint,
Marge requests,
package.json,
package-lock.json,
.env,

Scrum
Agile manifesto,
Jira,
User story,
Backlog,
Sprint,
Team velocity,

Backend:
SQL, NoSQL (Mongo), ORM, TCP/IP, DNS, HTTPS, LDAP, OAuth 2, Microservices, Redis, Kafka, Swagger Open API, MEAN, MERN, LAMP, CRON.